I n h a l t s v e r z e i c h n i s

		Seite
1.	Gerätebeschreibung	3
1.1.	Allgemeines	3
1.2.	Zentraleinheit	3
1.3.	Schreibwerk	5
1.4.	Tastaturen	5
2.	Programmierung	7
3.	Befehlsbeschreibung	7
3.1.	Leerbefehl	8
3.2.	Numerische Eingabe	8
3.3.	Numerische Ausgabe	9
3.3.1.	Numerische Ausgabe mit Druck	9
3.3.2.	Numerische Ausgabe mit Sicherheitsstern	12
3.4.	Zeichenausgabe	13
3.5.	Rechenbefehle	13
3.6.	Transportbefehle	17
3.7.	Verschiebebefehle	17
3.8.	Sprungbefehle	18
3.9.	Konstanten-Eingabe	22
3.10.	Adressensubstitution	25
3.11.	Sonstige Befehle	26
4.	Fehlerkontrollen	29
5.	Modulprogrammierung	29
5.1.	Prinzip der Modulprogrammierung	30
5.2.	Aufbau von Makroprogrammen auf der Gundlage der Modulprogrammierung	30

© VEB Robotron - Büromaschinenwerk Sömmerda 1980

		Seite
5.2.1.	Programmeingabe, Kopfprogramm KK	30
5.2.2.	Programm-Module	31
5.2.3.	Tabelle der Programm-Moduladressen KT	31
5.3.	Eingabe des Programmcodes	32
5.4.	Ergänzungen der Makrobefehlsliste	33
5.4.1.	ME - Modulende	33
5.4.2.	MAW - Modulauswahl	34
5.4.3.	MSP - Modulsprung	35
5.5.	Eingabe der Nebenbedingungen	36
5.5.1.	Nebenbedingung ACx	36
5.5.2.	Nebenbedingung Kapazität	37
5.5.3.	Nebenbedingung N-mal TAB	38
Anlage 1:	Befehlsliste	39

1. Gerätebeschreibung

1.1. Allgemeines

Die Basis des elektronischen Kleinfakturierautomaten (KFA) 1711 bildet eine elektrische Typenhebel-Schreibmaschine. Zur Durchführung von Fakturierarbeiten wurde dieses Schreibwerk um eine Zentraleinheit und um eine Zehner- bzw. Funktionstastatur erweitert.
Die Funktionsfähigkeit der Schreibmaschine wird durch diese Erweiterung nicht eingeschränkt.

1.2. Zentraleinheit

Die mikroprogrammgesteuerte Zentraleinheit des "KFA 1711" setzt sich aus folgenden Hauptbaugruppen zusammen:
- Steuereinheit
- Mikroprogrammspeicher
- Makroprogrammspeicher
- Datenspeicher

Das Mikroprogramm dient der Steuerung der Zentraleinheit und realisiert in seiner konkreten Form den Befehlsumfang der Makrobefehlsliste.
Der Mikroprogrammspeicher ist als Festwertspeicher auf der Basis von ROM-Halbleiterbausteinen aufgebaut und hat eine Kapazität von 2 K Byte.

Als Makroprogrammspeicher wird wahlweise ein Halbleiter-Festwertspeicher (ROM) oder ein programmierbarer Halbleiterspeicher (PROM) mit einer Kapazität von 0,25 K Byte eingesetzt. Dieser Speicher ist bis auf 0,5 K Byte aufrüstbar.

Die Unterbringung dieser Speicherschaltkreise erfolgt in einer Programmkassette, welche durch die Bedienkraft gewechselt werden kann.
Der Datenspeicher (RAM) hat eine Kapazität von 16 Worten zu je 15 Ziffern + Vorzeichen (64 Bit).
Da für interne Operationen 4 Sonderregister benötigt werden, stehen für die Anwenderprogramme 12 freie Speicherworte zur Verfügung. (vergl. Abb. 1)
Im Falle der Modulprogrammierung verringert sich die Zahl der freien Datenspeicherplätze um 2 Speicherworte.

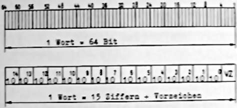


Abb. 1 Aufbau des Datenspeichers

Eine Speichersplittung kann softwareseitig (programmgesteuert) realisiert werden (s.Pkt. 3.9. Beispiel 3)
Die einzelnen Speicherworte werden folgendermaßen verwendet:

AC 0	Freie Datenspeicher
AC 1	
AC 2	
AC 3	
AC 4	
AC 5	
AC 6	
AC 7	
AC 8	
AC 9	
AC 10	Bei Verwendung der Modulprogrammierung als Programmregister belegt
AC 11	
AC 12	RR = Rechenregister
AC 13	ER = Eingaberegister
AC 14	MR = Register
AC 15	SR = Sonderregister

Abb. 2 Belegung des Datenspeichers

Das Rechenregister (AC 12) wird ausschließlich für Multiplikationen und Divisionen benötigt und kann als Zwischenspeicher verwendet werden. Die Programmierung von Multiplikationen und Divisionen mit AC 12 ist nicht möglich.

1.3. Schreibwerk

Als Ausgabeeinheit kommt ein Typenhebel-Schreibwerk mit folgenden Parametern zum Einsatz:

- Druckgeschwindigkeit	= ca. 12 Zeichen/s
- Wagenbreite	= 320/460 mm
- druckbare Zeichen/Zeile	= 117/167
- Zeichenteilung	= 2,6 mm
- Zeilenabstand (1-zeilig)	= 4,25 mm
- Zeichenvorrat	= 92 Zeichen (Groß-/Kleinbuchstaben)
- Durchschläge	= max. 1o (abhängig von der Papierqualität)

Für die Formulartechnik ist ausschließlich Einzelformularverarbeitung vorgesehen.

1.4. Tastaturen

Die Eingabe aller zu druckenden alpha-numerischen Informationen erfolgt über die Tastatur des Schreibwerkes. Bestandteil dieses Schreibwerkes sind alle Tasten, die zur Bedienung erforderlich sind (WRZ, Tabulator, Zeilenschaltung, usw.). Für die Eingabe aller zu verrechnenden Informationen sowie zur Auslösung bestimmter Maschinenfunktionen steht eine Zusatztastatur (Zehner- und Funktionstastatur) zur Verfügung (s. Abb. 3)

Neben der √-Tastatur befindet sich eine Taste zum Abschalten des Schreibwerkmotors.

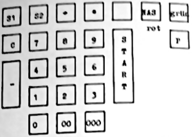


Abb. 3

Zehner- und Funktionstastatur des KFA 1/11

Die einzelnen Tasten haben dabei folgende Bedeutung:

0, 1 ... 9	=	Internationale Zehnertastatur mit Fühlpunkt auf der Taste 5 für Blindbedienung
START	=	Taste zur Auslösung des Transportes aller in das Eingaberegister eingegebenen Ziffern (VZ) zum Datenspeicher in Verbindung mit dem entsprechenden Makrobefehl
-	=	Eingabe des Minuszeichens in das Eingaberegister
C	=	Korrektur (Löschung) des Eingaberegisters bei Falscheingaben vor START-Auslösung
•	=	Start • Eingabeüberführung in Verbindung mit Programmbefehl und gleichzeitige Einschaltung des Selektors T
•	=	Start • Eingabeüberführung in Verbindung mit Programmbefehl und gleichzeitige Einschaltung des Selektors Z
S 1	=	Taste zum Setzen bzw. Löschen von Selektor 1
S 2	=	Taste zum Setzen bzw. Löschen von Selektor 2
P	=	Taste zur Einstellung des Grundzustandes für das Makroprogramm (Start des Makroprogrammes mit dem Befehl auf der Makroadresse ØØ)
grün	=	Anzeige für manuelle numerische Eingabe bzw. Anzeige für Kapazitätsüberschreitung (blinkend)
rot	=	Anzeige Netzausfall

2. Programmierung
Die Programmierung des KFA "daro 1711" erfolgt in einer wagenschrittunabhängigen Form. Zur Programmspeicherung sind drei Varianten vorgesehen:

a) Speicherung der länderspezifischen Standardprogramme auf Halbleiter-Festwertspeichern (max. 512 Byte), freie Speicherplätze AC Ø ... AC 11

b) Speicherung der kundenspezifischen Anwenderprogramme auf programmierbaren Halbleiterspeichern (max. 512 Byte), freie Speicherplätze AC Ø ... AC 11

c) Speicherung der länder- oder kundenspezifischen Programmodule auf Halbleiter-Festwertspeichern (max. 512 Byte) sowie Eingabe des Programmcodes in den RAM-Speicher AC 10 (max. 15 Tetraden) und Eingabe einer Nebenbedingung pro Modul in den RAM-Speicher AC 11 (max. 15 Tetraden);
damit stehen für die Modulprogrammierung die Speicherplätze AC Ø ... AC 9 als freie Speicher zur Verfügung.

Die einzelnen Makrobefehle haben eine variable Befehlslänge. Dabei benötigen 70 % aller Befehle 1 Byte und 30 % der Befehle 2 Byte.
Somit wird eine maximale Ausnutzung der Programmspeicherkapazität gewährleistet.

3. Befehlsbeschreibung
Zur Programmierung der Anwenderprogramme stehen die in der Anlage 1 (Befehlsliste) aufgeführten Makrobefehle zur Verfügung. Dabei spielen die Makrobefehle MS, MAW und MSP nur im Zusammenhang mit der Modulprogrammierung eine Rolle. Sie werden daher unter Punkt 6. beschrieben.

Die einzelnen Befehle haben dabei folgende Bedeutung:

3.1. Leerbefehl

Als Leerbefehl können folgende Codierungen verwendet werden:

KO keine Operation, Leerbefehl Ø Ø
f f

Der Leerbefehl löst bei der Abarbeitung keine Operation aus und bewirkt die Erhöhung des Makrobefehlszählers um +1.
Der KO-Befehl kann verwendet werden, um Stellen im Makroprogrammspeicher für das spätere Nachtragen von Befehlen freizuhalten (Codierung = Ø Ø) bzw. um falsche Makrobefehle unwirksam zu machen (Codierung = ff).

3.2. Numerische Eingabe

Die manuelle Eingabe von numerischen Daten erfolgt über die Zehnertastatur in das Eingabe-Register.
Bei Betätigung der START-Taste (bzw. Start *, Start °) werden die eingegebenen Daten mit Hilfe des Befehls

EZ	Numerische Eingabe: ER → ACx	e	ACx

in den Datenspeicher überführt.
Die Kennzeichnung negativer Werte erfolgt durch Drücken der Minus-Taste vor der Startauslösung.
Bei fehlerhafter Eingabe besteht noch vor START-Auslösung die Möglichkeit, mit Hilfe der C-Taste das Eingabe-Register zu löschen.
Zur Beschleunigung des Programmablaufes kann mit überlappter Eingabe und Startvorwahl gearbeitet werden.
Die Starttasten ° und * haben keinen Einfluß auf Stellenabstreichung, Speicherlöschung und Abdruck von Sonderzeichen (*, °).

Zur Eingabe von Alphatext über die Schreibmaschinen-Tastatur ist die Programmabarbeitung ebenfalls mit dem EZ-Befehl zu unterbrechen. Die Fortsetzung des Makroprogrammes erfolgt dann mit einer Starttaste. Bei ist zu beachten, daß die jeweils programmierte Eingabeüberführung (ER ⟶ ACx) ausgeführt wird.

3.3. Numerische Ausgabe

Zur Druckausgabe von numerischen Informationen aus dem Datenspeicher werden folgende Makrobefehle verwendet:

3.3.1.

ANP	Numerische Ausgabe: ACx ⟶ Druck	? ACx	PFKo	Kap.
		a	b	c

Dieser 2-Byte-Befehl enthält neben dem Befehlscode folgende Parameter:

a) ACx = Adresse des Speicherplatzes, dessen Inhalt auszudrucken ist. Die Verschlüsselung der Speicheradressen erfolgt in hexadezimaler Form.

b) Behandlung der führenden Stellen und der Dezimalstellen-Abtrennung:

P = Ø ⟶ Dezimalstellenabtrennung durch Komma
P = L ⟶ Dezimalstellenabtrennung durch Punkt

F = Ø ⟶ Druck führender Stellen als Null
F = L ⟶ Unterdrückung der führenden Stellen (Zwischenraum)

Ko = ØØ ⟶ keine Dezimalstellenabtrennung
ØL ⟶ 1 Dezimalstelle nach Komma (Punkt)
LØ ⟶ 2 Dezimalstellen nach Komma (Punkt)
LL ⟶ 3 Dezimalstellen nach Komma (Punkt).

c) Zur Codierung dieser Tetrade kann folgende Tabelle verwendet werden (Abb. 4):

führende Stellen / Dez.-Stellen-Abtrennung	Null	Z W R
durch Komma		
0 Stellen	∅	4
1 Stelle	1	5
2 Stellen	2	6
3 Stellen	3	7
durch Punkt		
0 Stellen	∅ (8)	4 (c)
1 Stelle	9	d
2 Stellen	a	e
3 Stellen	b	f

Abb. 4: Codierungstabelle für führende Stellen und Dezimalstellenabtrennung

c) Kap = Ausgabekapazität
Die maximale Anzahl der in dieser Spalte auszugebenden Ziffern ist in dieser Tetrade hexadezimal verschlüsselt einzutragen:

∅ Stellen	=	∅
1 Stelle	=	1
2 Stellen	=	2
3 Stellen	=	3
4 Stellen	=	4
5 Stellen	=	5
6 Stellen	=	6
7 Stellen	=	7
8 Stellen	=	8
9 Stellen	=	9
10 Stellen	=	a
11 Stellen	=	b
12 Stellen	=	c

13 Stellen • d
14 Stellen • e
15 Stellen • f

Bei der Ausgabe negativer Werte wird das Minuszeichen automatisch unmittelbar nach der letzten Ziffer gedruckt. Bei positiven Werten wird an dieser Stelle ein Zwischenraum ausgeführt.
Wenn die Stelligkeit des auszudruckenden Speicherplatzes größer ist, als die jeweils programmierte Ausgabekapazität, dann wird der Selektor F eingeschaltet, die Ausgabe unterdrückt und die Kapazitätsüberschreitung optisch angezeigt (blinkende Lampe).
Nach einer Kapazitätsüberschreitung ist die C-Taste zu drücken, anschließend kann die Eingabe wiederholt werden.

Beispiel:

Art.-Nr.	T e x t	Menge
99999		99,99
12345	Butter	10,00
34567	Milch T	20,00

```
10  EZ, 14      ee     (Eingabe: Art.-Nr.)
11  AND, 14     7e     (Druck:   Art.-Nr.)
12  (N Ø,5)     Ø5
13  SPF         6e
14  (10)        Øa
15  EZ, Ø       eØ     (Eingabe/Druck: Text, Eingabe: Menge)
16  TAB         f2
17  AND, Ø      7Ø     (Druck: Menge)
18  (Z 2,4)     64
19  SPF         6e
20  (15)        Øf
21
     :
```

Befehle TAB siehe Pkt. 4.7.
SPF siehe Pkt. 4.1Ø.

3.3.2.

AND *	num.Ausgabe mit Sicherheitsstern	5 ACx	P1Ko	Kap

Dieser Ausgabefehl unterscheidet sich vom oben beschriebenen AND-Befehl nur dadurch, daß die führenden Stellen stets als * angegeben werden.
Der Unterschied in der Programmierung besteht im Vergleich zum AND-Befehl darin, daß der Operationscode (1. Byte, 2. Tetrade) mit 5 codiert wird und das Bit F (2. Byte, 2. Tetrade) 1 gesetzt wird.
Die Programmierung bezüglich Interpunktionszeichen, Dezimalstellen, Kapazität und Ausgabespeicher ist wie unter 3.3.1. beschrieben vorzunehmen.
Zu beachten ist, daß alle 2-Byte-Befehle generell zwei Speicherplätze im Makroprogrammspeicher belegen.

3.4. Zeichenausgabe

Beim KFA 1711 besteht die Möglichkeit, mit Hilfe der nachfolgenden Makrobefehle 7 verschiedene Sonderzeichen abzudrucken:

AMD (%)	Ausgabe Zeichen:	%	f	c
AMD (‰)	Ausgabe Zeichen:	‰	f	d
AMD (°)	Ausgabe Zeichen:	°	f	e
AMD (*)	Ausgabe Zeichen:	*	f	5
AMD (Pkt)	Ausgabe Zeichen:	Punkt	f	b
AMD (ZWR)	Ausgabe Zeichen:	Zwischenr.	f	1
AMD (-)	Ausgabe Zeichen:	Minus	f	a

Bei der Ausführung dieser Befehle wird das jeweilige Zeichen an die entsprechende Druckstelle abgedruckt.

3.5. Rechenbefehle

Zur Ausführung der erforderlichen Rechenfunktionen stehen folgende Makrobefehle zur Verfügung:

ADV	Addition: ACx + MD --> ACx	a	ACx	-	-
AD1V	Addition: ACx + 1 --> ACx	9	ACx	-	-
SUV	Subtraktion: ACx - MD --> ACx	b	ACx	-	-
MU,N	Multiplikation: ACx x MD --> MD (N= Stellen-Versch.)	5	ACx	1	N
DI,N	Division: MD : ACx --> MD (N= Stellen-Versch.)	5	ACx	Ø	N

Alle Rechenoperationen werden vorzeichengerecht ausgeführt. Der Inhalt des Registers MD (bei Addition bzw. Subtraktion) und der Inhalt des adressierten Speicherplatzes ACx (bei Multiplikation und Division) bleibt unverändert erhalten. Die Programmierung des Befehles ADV,14 ist möglich und bewirkt eine Verdopplung des Inhalts von MD.
Nicht erlaubt ist die Programmierung des Befehles SUV,14 (MD - MD).

Bei der Programmierung des Befehls AD1V (ACx + 1) ist darauf zu achten, daß der Inhalt von ACx nicht den Wert Ø annehmen darf. Eine Ø-in ACx kann nur durch folgende Operationen entstehen:

- Manuelle Eingabe einer-Ø über die Zehnertastatur;
- Konstanteneingabe eines Vorzeichens (TRO,14/TKON(a);
- Rechts- bzw. Linksverschiebungen in MD, wenn <MD>
- negativ ist.

Bei der Multiplikation wird das errechnete Produkt generell gerundet. Die erforderliche Stellenabstreichung ist jeweils vom Programmierer zu ermitteln, wobei folgende Formel verwendet werden kann:

$$N = n_{Md} + n_{Mr} - n_{Pr}$$

N = zu programmierende Stellenabstreichung
n = Stellen nach dem Komma
Md = Multiplikand (MD)
Mr = Multiplikator (ACx)
Pr = Produkt

z.B.:

- 12,<u>000</u> x 10,<u>00</u> = 120,<u>00</u>
  3 + 2 - 2 → N = 3

- 100,<u>00</u> x 500 = 50000,00
  2 + 0 - 2 → N = Ø

Die Anzahl der abzustreichenden Stellen wird im 2.Byte des Befehls hexadezimal eingetragen (N = 3 → Ø3: N = 1Ø → Øa).

Zu beachten ist, daß einer der beiden Faktoren (ACx oder MD) nur maximal 14 Stellen sein darf.

Wird eine Multiplikation ohne Rundung benötigt, dann muß die Stellenabstreichung durch eine Rechtsverschiebung realisiert werden (sh. Pkt. 3.7.).

Bei der Multiplikation ist dabei N = Ø zu programmieren. Nicht zulässig ist die Programmierung des Befehls MU,14 (MD x MD).
Auch die Divisionsbefehle werden mit Rundung ausgeführt. Die erforderliche Stellenverschiebung kann vom Programmierer nach folgender Formel ermittelt werden:

$$N = n_{Qu} + n_{Dr} - n_{Dd} + 3$$

N = zu programmierende Stellenverschiebung
n = Stellen nach dem Komma
Qu = Quotient
Dr = Divisor
Dd = Dividend

Eine Division mit dem Befehl DI,14 (MD : MD) ist nicht zugelassen.

z.B.:

160,<u>00</u> : 4,<u>00</u> = 40,<u>00</u>
-2 + 2 + 2 + 3 ⟶ N = 5

300,<u>00</u> : 6 = 50,<u>0</u>
-2 + Ø + 1 + 3 ⟶ N = 2

Wird eine Division ohne Rundung benötigt, dann ist die erforderliche Stellenverschiebung durch Verschiebebefehle zu realisieren.
Bei der Ausführung einer Division mit dem Divisor Ø wird ohne Fehlermeldung mit dem Quotienten = Ø weitergerechnet.

Beispiel $\frac{(A - B)\,C}{D} = E$

A	B	C	D	E
999999	999999	99,99	9999,9	999999.99 •
50000	40000	2,00	5000,0	40.00
8000	10000	3,00	20,0	300.00-

```
18  EZ,Ø       e Ø
    AND,Ø      7 Ø
20  (ZØ,6)     4 6
    SPF        6 e
    (18)       1 2
    EZ,14      e e
24  AND,14     7 e
    (ZØ,6)     4 6
    SPF        6 e
    (23)       1 7
28  SUV,Ø      b Ø
    EZ,14      e e
    AND,14     7 e
    (Z2,4)     6 4
32  SPF        6 e
    (29)       1 d
    MU,Ø       4 Ø
    (ØØ)       Ø Ø
36  EZ,Ø       e Ø
    AND,Ø      7 Ø
    (Z1,5)     5 5
    SPF        6 e
4Ø  (36)       2 4
    DI,Ø       5 Ø
    (Ø4)       Ø 4
    AND,14     7 e
44  (PZ2,8)    e 8
```

Beispiel: Multiplikation ohne Rundung

Menge	Preis	Betrag
999,9	99,99	99999,99
10,2	3,99	40,69 T

```
EZ,14          VR,1
AND,14         TAB
(Z1,4)         AND,14
SPF            (Z2,7)
(...)          SPF
EZ,Ø           (...)
AND,Ø            ⋮
(Z2,4)
SPF
(...)
MU,Ø
```

3.6. Transportbefehle

Folgende Transportbefehle stehen dem Programmierer des KFA 1711 zur Verfügung:

TR	MD ⟶ ACx	3	ACx
TRV	ACx ⟶ MD	8	ACx
TRØ	ACx: Ø ⟶ ACx	2	ACx

Mit der Ausführung eines Transportbefehles wird jeweils der Inhalt eines Wortes (64 Bit, einschließlich Vorzeichen) in das adressierte Register überführt.
Der Inhalt des 1. Registers bleibt dabei unverändert erhalten.
Mit dem Befehl TRØ wird eine Einzellöschung des adressierten Speicherplatzes (einschließlich Vorzeichen) erreicht.

3.7. Verschiebebefehle

Mit den Befehlen

VL,N	(N Stellen) in MD	Ø	N
VR,N	(N Stellen) in MD N ≠ Ø	1	N

können im Register MD (AC14) Rechts- bzw. Linksverschiebungen um maximal 15 Stellen (N) vorgenommen werden.
Das Vorzeichen in MD bleibt bei beiden Verschiebeoperationen unverändert erhalten.
Alle aus dem Zahlenbereich verschobenen Ziffern gehen verloren; die freien Stellen werden mit Nullen aufgefüllt.
Die Rechtsverschiebung wird ohne Rundung ausgeführt.
Der Befehl VL,Ø bewirkt keine Veränderung des Inhalts von MD, er ist mit dem Leerbefehl ØV identisch, während VR,Ø nicht verwendet werden darf.

3.8. Sprungbefehle

Zur Programmierung von Programmverzweigungen und logischen Entscheidungen stehen folgende Sprungbefehle zur Verfügung:

SPU	Unbedingter Sprung	6	Ø(1)		
SPM	Bedingter Sprung, wenn MD < Ø	6	2(3)		
SPV	Bedingter Sprung, wenn MD ≮ Ø	6	4(5)		
SP1	Bedingter Sprung bei Selektor 1	6	6(7)	Sprung-adresse	
SP2	Bedingter Sprung bei Selektor 2	6	8(9)		
SPT	Bedingter Sprung bei Start *	6	a(b)		
SPZ	Bedingter Sprung bei Start °	6	c(d)		
SPF	Bedingter Sprung bei Selektor F (Kap.-Überschreitung)	6	e(f)		

Bei den Sprungbefehlen unterscheidet man zwischen unbedingten und bedingten Sprüngen. Während die unbedingten Sprünge immer ausgeführt werden, erfolgt die Ausführung der bedingten Sprünge nur dann, wenn die jeweilige Sprungbedingung erfüllt ist.
Bei der Befehlscodierung wird im 2. Byte des Befehls die Sprungadresse hexadezimal verschlüsselt eingetragen.
Für die Verwendung der Adressen 256 bis511 (2.Schaltkreis) ist im 1. Byte des Befehls Bit 1 = L zu programmieren (Klammerwerte):

```
z.B.: SPU,29     (6Ø)
       29        (1d)
      SPU, 289   (61)
       289       (21).
```

Die bedingten Sprungbefehle lassen sich in drei Gruppen einteilen:

1. Starttastenabhängige Sprungbefehle
   Die Einschaltung der jeweiligen Sprungbedingung erfolgt durch Betätigung der entsprechenden Starttaste:

- START = Es wird kein Selektor eingeschaltet;
- Start* = Einschaltung des Selektors T
  (Abfrage mit SPT-Befehl);
- Start° = Einschaltung des Selektors Z
  (Abfrage mit SPZ-Befehl);

Die eingeschaltete Sprungbedingung bleibt bis zur nächsten Eingabestelle (EZ-Befehl) erhalten, auch wenn in der Zwischenzeit durch überlappte Eingabe bereits die nächste Eingabe vorgenommen wurde.

```
z.B.:   15   EZ,14      e e
        16   SPT        6 a   ————→ Rechnungsab-
        17   (59)       3 b           schluß
        18   SPZ        6 d   ————→ Zwischen-
        19   (399)      8 f           summe
        20   AND,14     7 e
        21   (NØ,6)     Ø 6
        22   SPF        6 e
        23   (15)       Ø f
              ↓
             Artikel-Zeile
```

2. Selektorabhängige Sprungbefehle

Selektoren sind elektronisch gespeicherte Marken, welche durch Tasten (S1, S2) gesetzt bzw. gelöscht werden können und die durch eine Abfrage mittels eines Sprungbefehls (SP1, SP2) zu Programmverzweigungen verwendet werden können.
Das manuelle Setzen bzw. Löschen der Selektoren S1 und S2 ist nur an einer Eingabestelle (EZ) möglich.
Der Selektor F wird automatisch bei einer Kapazitätsüberschreitung gesetzt. Mit der Ausführung des SPF-Befehls wird der Selektor F gelöscht(vergl. Pkt. 5).

z.B.: Ausgabe einer Preisspalte mit unterschiedlichen Bezugsmengen

Menge	Preis	Betrag
9999	99,99	99999,99 *
2000	20,00 ‰	40,00
50	5,00	250,00
700	12,00 %	84,00

0	TRO,6	2	6
	EZ,0	e	0
	AND,0	7	0
	(Z0,4)	4	4
4	SPF	6	e
	(o1)	0	1
	EZ,14	e	e
	AND,14	7	e
8	(Z2,4)	6	4
	SPF	6	e
	(6)	0	6
	SP1	6	6
12	(20)	1	4
	SP2	6	8
	(25)	1	9
	AMD/ZWR	f	1
16	MU (0)	4	0
	(00)	0	0
	SPU	6	0
	(28)	1	c
20	AMD %	f	c
	MU,0	4	0
	(02)	0	2
	SPU	6	0
24	(28)	1	c
	AMD ‰	F	d
	MU,0	4	0
	(03)	0	3
28	AND,14	e	e
	(Z2,7)	6	7
	SPF	6	e
	(01)	0	1
32	ADV,6	a	6
	WRZ	f	3
	SPU	6	0
	(5)	0	5

Sel. 1 = %

Sel. 2 = ‰

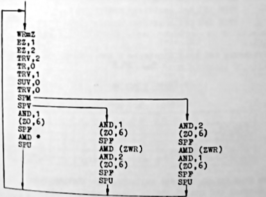
3. Von logischen Vergleichen abhängige Sprungbefehle

Die Ausführung der Befehle SPM und SPV ist abhängig vom Inhalt des Registers MD (= AC14).
Der SPM-Befehl wirkt nur dann, wenn der Inhalt von $MD < 0$ ist.
Die Bedingung für den SPV-Befehl ist nur dann erfüllt, wenn der Inhalt von $MD \neq 0$ ($MD < 0$ oder $MD > 0$) ist.

z.B.: Die eingegebenen Zahlen (AC1 und AC2) sind in aufsteigender Folge auszudrucken. Bei Gleichheit erfolgt nur ein Ausdruck in Spalte A, welcher durch * zu kennzeichnen ist.

Eingaben: 120 / 190
240 / 155
265 / 265

A	B
999999	999999
120	190
155	240
265*	



3.4. Konstanten-Eingabe

Zum Aufbau von konstanten Zahlen kann folgender Befehl verwendet werden:

TKON,X	Konstanten-Eingabe: KON ——→ MD (VZ-Stelle)	[illegible]

Mit diesem Befehl kann eine einstellige konstante Ziffer auf die Vorzeichenstelle von MD ([illegible]) eingetragen werden.
Bei der Ausführung dieses Befehls werden durch das Mikroprogramm folgende Funktionen realisiert:

1. Linksverschiebung des Inhalts von MD um 1 Stelle (einschließlich Vorzeichenstelle ; ohne 15. Stelle)
2. Eintragen der im Befehl eingestellten 4-Bit-Konstante in die Vorzeichenstelle von MD.

Es macht sich demzufolge erforderlich, als letztes das Vorzeichen mit dem TKON-Befehl einzutragen:

TKON (0) bei positiver Zahl
TKON (a) bei negativer Zahl.

Beispiel:
Zur Berechnung ist eine Konstante ([illegible]) in MD aufzubauen.

Inhalt von MD

Stelle:	5.	4.	3.	2.	1.	VZ
:	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
TKO,14	0	0	0	0	0	[illegible]
TKON (2)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
TKON (6)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
TKON (a) (für VZ)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible] -
VZ,2	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible] -

Mit dem TKON-Befehl besteht weiterhin die Möglichkeit Sonderzeichen auf die VZ-Stelle einzutragen, welche dann mit dem Ausgabebefehl abgedruckt werden können.

Folgende Sonderzeichen können verwendet werden:

TKON (Ø) = Zwischenraum (= positives VZ)
TKON (1) = 1
⋮ ⋮
TKON (9) = 9
TKON (a) = Minus
TKON (b) = Punkt
TKON (c) = %
TKON (d) = ‰
TKON (e) = °
TKON (f) = *.

Dabei ist zu beachten, daß bei Rechenoperationen auf der Vorzeichenstelle außer Zwischenraum (Ø) und Minus (a) keine Sonderzeichen stehen dürfen.

Beispiel 2:

Der konstante Steuersatz (=16,20 %) ist auszudrucken und in AC5 für weitere Berechnungen zu speichern.

Variante 1:

			5	4	3	2	1	VZ
TRO,14	2	e	0	0	0	0	0	0
TKON (1)	d	1	0	0	0	0	0	1
TKON(6)	d	6	0	0	0	0	1	6
TKON(2)	d	2	0	0	0	1	6	2
TKON(Ø)	d	Ø	0	0	1	6	2	0
VL,1	Ø	1	0	1	6	2	0	0
TR,5	3	5	0	1	6	2	0	0
TKON(%)	d	c	1	6	2	0	0	c
VR,1	1	1	0	1	6	2	0	c
AND,14	7	e	0	1	6	2	0	c
Z2,4	6	4	0	1	6	2	0	c

Inhalt von MD

Variante 2:

TRO,14
TKON(1)
TkoN(6)
TKON(2)
TKON(O)
TKON(O) (oder VL,1)
TR,5
AND,14
(Z2,4)
AMD(%)

Druckbild
16,20%

ohne ZWR

Druckbild
16,20 %

mit ZWR

__Beispiel 3:__ Speichersplittung

In AC6 ist sowohl die Rechnungs-Nr. als auch das Rechnungsdatum zu speichern.

```
<AC6> =   | 1 | 9 | 1 | 2 | 7 | 7 | 0 | 0 | 1 | 2 | 4 |   | + |
                      Datum              Rechn.-Nr
```

(Eingabe: Datum + Re.-Nr./START)

```
EZ,6
  :
WRZ
TAB
TRV,6
VL,10
VR,10
AND,14        (Druck: Re.-Nr.)
(NØ,5)
AD1V.6        (Re.-Nr. + 1)
TAB
TRV,6
VR,9
TKON(b)
VR,1
AND,14        (Druck: ZWR 19.)
(P,N,Ø,2)
TRV,6
VL,6
VR,11
AND,14        (Druck:  12.77)
(P,N,2,4)
  :
```

oder:

```
EZ,14     Datum
VL,5
TR,6
EZ,14     Re-Nr.
ADV,6
  :
WRZ
```

3.10. Adressensubstitution

Durch die Programmierung der Adressensubstitution mit dem Befehl

AS	Adressensubstitution	f	9

wird bei jedem nachfolgenden AC-Befehl der Adressteil (ACr) mit dem Inhalt des jeweils adressierten Speicherplatzes ausgetauscht. Das heißt, der Inhalt des adressierten Wortes wird zur neuen Speicheradresse.
Der AS-Befehl muß unmittelbar vor dem jeweiligen AC-Befehl programmiert werden.

Z.B. Der Betrag ist für die einzelnen Artikelgruppen 1 – 6 (werthöchste Stelle der Artikel-Nr.) zu summieren

Art.-Nr.	Menge	Preis	Betrag
99999	9999	99,99	999999,99 *
12345	10	5,00	50,00
23608	15	12,00	180,00

```
WRmZ
RZ, 14   (Art.-Nr.)
AND, 14
(NØ5)
SPF
VR, 4
TR, Ø
RZ, 14   (Menge)
AND, 14
(ZØ4)
SPF
```

```
RZ, 11   (Preis)
AND, 11
(Z2,4)
SPF
MU, 11
(ØØ)
AND, 14  (Betrag)
(Z2,8)
SPF
ADV, 10
AS
ADV, 0
SPU
```

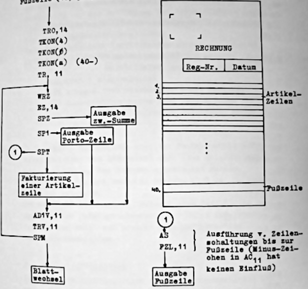
3.11. Sonstige Befehle

Zur Steuerung der Schreibwerkfunktionen vom Makroprogramm und zum Löschen/Setzen der Selektoren sind folgende Befehle vorgesehen:

WRZ	Wagenrücklauf mit Zeilenschaltung	f	3		
WRoZ	Wagenrücklauf ohne Zeilenschaltung	5	0	3	0
PZD,N	Zeilenschaltung (N-zeilig)	c	N / ACx		
TAB	Tabulator-Sprung	f	2		
TAB,N	Tabulator-Sprung ·	5	0	2	N
ZLBS	Löschen Selektor 1 + 2	f	7		
ZS 1	Setzen Selektor 1	f	4		
ZS 2	Setzen Selektor 2	f	8		

Der Befehl WRZ bewirkt einen Wagenrücklauf mit Zeilenschaltung bis zum eingestellten linken Randsteller. Der Befehl WROZ bewirkt einen Wagenrücklauf bis zum linken Rand ohne Zeilenschaltung. Voraussetzung dafür ist, daß mindestens 3 Schrittstellen bis zum linken Rand garantiert sind. Sind die 4 Schrittstellen nicht garantiert, so muß dem Befehl WROZ ein Befehl TAB voran gesetzt werden.
Mit dem Befehl PZL,N können Zeilenschaltungen ohne Wagenbewegung programmiert werden. Die Anzahl der Zeilenschaltungen (=N) ist im Befehl hexadezimal zu verschlüsseln. Mit einem Befehl können somit max. 15 Zeilenschaltungen (cf) ausgeführt werden.
Wird vor dem Befehl Zeilenschaltung der Befehl AS - (Adreß-Substitution) programmiert, dann wird der Inhalt der 1.Tetrade des PZL-Befehls als Speicheradresse (ACx) interpretiert. Die Anzahl der auszuführenden Zeilenschaltungen wird dabei durch den Inhalt dieses Speicherplatzes bestimmt.

Beispiel:
Bei einer Faktura (s. Skizze) ist nach der letzten Artikelzeile die Anzahl der Zeilenschaltungen bis zur Fußzeile (40) programmabhängig zu bestimmen.



Mit dem TAB-Befehl wird eine Vorwärts-Tabulation bis zur nächsten Tabulator-Stop-Stelle ausgeführt. Der Tabulator-Stop ist manuell durch die entsprechende Schreibwerk-Taste an der Stelle des ersten zu druckenden Zeichens zu setzen.

Der TABN-Befehl bewirkt die N-malige Ausführung des TAB-Befehls. Es gilt dabei $1 \leqq N \leqq 15$. Der Fall N = ∅ spielt nur bei Anwendung der Modulprogrammierung eine Rolle. Er wird unter Abschnitt 5.5.3. abgehandelt.
Mit den Befehlen ZS1 und ZS2 können die Selektoren 1 und 2 an jeder beliebigen Stelle programmabhängig eingeschaltet werden.
Das Löschen beider Selektoren erfolgt mit dem Befehl ZLÖS.

Beispiel: Ausgabe von Rechnungsdatum und Fälligkeitsdatum mit einer Befehlsfolge

```
EZ,6     (Eingabe Rechn.-Datum)

ZLÖS
TRV,6
TR,7
TRV,7
TKON (b)
VR,3
AND,14
(PN2,4)
TRV,7
VL,13
VR,13
AND, 14
(N2,2)
TAB
TAB
SP1
EZ,7     (Eingabe Fälligkeitsdatum)
ZS1
SPU
```

4. Fehlerkontrollen

Um eine relativ hohe Sicherheit bei der Verarbeitung der numerischen Informationen zu gewährleisten und um Eingabefehler in bestimmten Grenzen ausschließen zu können, wird vom KPA robotron 1711 eine Kontrolle der Ausgabekapazität durchgeführt.
Dabei wird vor der Ausführung des Ausgabebefehls (AND) die Anzahl der auszugebenden Ziffern mit der programmierten Ausgabekapazität verglichen.
Bei einer erkannten Kapazitätsüberschreitung (Anzahl Ziffern in AUx > Ausgabekapazität) wird

- Selektor F eingeschaltet
- die Ausgabe verhindert
- die Makroprogrammabarbeitung unterbrochen und
- der Fehler optisch angezeigt (blinkende grüne Lampe)

Nach Drücken der C-Taste wird der Fehlerzustand aufgehoben und über den SPP-Befehl kann eine Eingabestelle zur Wiederholung der Eingabe angesteuert werden. Unmittelbar vor der nächsten Ausgabe wird der Selektor F wieder gelöscht.
Kurzzeitiger Netzausfall, welcher eine Veränderung des Inhaltes der Datenspeicherworte zur Folge hat, wird optisch angezeigt und bewirkt eine Unterbrechung des Programmablaufes.

5. Modulprogrammierung

Zur weiteren Erhöhung einer flexiblen Anwendung des KPA robotron 1711 wurdedie Möglichkeit der Modulprogrammierung geschaffen. Diese Programmierart verlangt keine besondere Zusatzeinrichtung, sondern ist vielmehr neben der schon beschriebenen Programmierung in jedem Gerät möglich.

5.1. Prinzip der Modulprogrammierung

Im Unterschied zur herkömmlichen Programmierweise werden bei Anwendung der Modulprogrammierung keine kompletten Makroprogramme mehr aufgestellt. Im Makroprogrammspeicher stehen dagegen eine bestimmte Anzahl von Programmteilen (maximal 16 verschiedene), die im weiteren als Module bezeichnet werden. Jeden Modul einer Kassette wird definitiv eine Modulnummer N mit 0 ≤ N ≤ 15 zugeordnet.
Durch die Eingabe von maximal 15 Modulnummern über die Zehnertastatur des Gerätes wird das benötigte Makroprogramm zusammengestellt. Dieser eingegebene Programmcode gibt also an, welche Module in welcher Reihenfolge abgearbeitet werden sollen. Der Programmcode steht im AC 1Ø. Er wird erst verändert, wenn ein neuer Programmcode eingegeben oder die Maschine vom Netz getrennt wird.

5.2. Aufbau von Makroprogrammen auf der Grundlage der Modulprogrammierung

Die modular aufgebauten Makrokassetten haben prinzipiell folgende Struktur

5.2.1. Programmeingabe, Kopfprogramm KK

```
        EZ,13
        SPT   KK Ø 1
        ME
KKØ1    WRZ
        EZ,10
        WRZ
        EZ,11
        ME
```

Der Befehl EZ,13 hat hierbei zwei Funktionen:

- Er weist die Maschine auf die Anwendung der Modulprogrammierung hin (nur auf der Makroadresse ØØ).

- Er unterbricht die Maschine nach dem Drücken der P-Taste, um der Bedienkraft die Neueingabe eines Programmschlüssels (Start * ⟶ Sprung zu KKØ 1) oder die Fortsetzung des gespeicherten Programms zu ermöglichen (START).

Der Befehl EZ,1Ø bewirkt die Eingabe des Programmschlüssels (mit Kontrollausgabe)
Der Befehl EZ,11 bewirkt die Eingabe einer Nebenbedingung pro Modul (Ausgabekapazität, ACx, N bei TABN Befehl). Die WEZ-Befehle gestatten ein spaltengerechtes Ausdrucken des Programmcodes und der dazugehörigen Nebenbedingungen.

5.2.2. Programm-Module

Nach dem Kopfprogramm KK schließen sich in beliebiger Reihenfolge die einzelnen Module an. Alle Stellen, an denen ein Modul verlassen wird, sind durch einen ME (Modulende)-Befehl zu markieren.

In diesem Speicherbereich, der im allgemeinen die Makroadresse Ø1 DF nicht überschreiten darf, können maximal 16 verschiedene Module stehen.

5.2.3. Tabelle der Programm-Moduladressen KT

Der Makroprogrammteil KT umfaßt den Bereich der Makroadressen Ø1 EØ bis Ø1 FF. Hier sind unbedingte Sprünge zu den Anfängen der einzelnen Modulen abgespeichert.

Ordnet man jedem Modul eine hexadezimale Ziffer n mit ∅ ≤ n ≤ f zu, so sind alle Module mit ungeradem n durch Sprungbefehle auf den Adressen ∅1 E∅, ∅1 E2, ∅1 E4, ..., ∅1 EE anzuwählen. Die Sprünge zu Modulen mit geradem n sind auf den Adressen ∅ 1 F∅, ∅1 F2, ..., ∅ 1 FE abzuspeichern.

Beispiel:	Modul ∅	auf Speicheradresse	
	(Beginn ab Adresse ∅1 34)	∅1 F∅:	61
		∅1 F1:	34
	Modul 3	∅1 E2:	6∅
	(Beginn ab Adresse ∅∅ 1b)	∅1 E3:	1b

5.3. Eingabe des Programmcodes

Die im KK-Programm (Abschnitt 5.2.1.) stehenden Eingabebefehle EZ 1∅ und EZ 11 dienen der Eingabe des Programmcodes sowie der offenen Nebenbedingungen für die einzelnen Modulen über die Zehnertastatur des KPA. Da diese die Hex-Zeichen A, B, ..., F nicht enthält, werden an dieser Programmstelle alle Hex-Ziffern dezimal zweistellig eingegeben (z.B. C = 12). Nach der Eingabe der Ziffer eines Hex-Zeichens (i.o. Beispiel die12) erfolgt der Ausdruck in dezimaler Form mit Kommaabtrennung (z.B. 12,). Gleichzeitig wird mit der Eingabe jedes Hex-Zeichens ein Programmcodezähler angesprochen und um 1 weitergezählt.

Beispiel 1: Es wurde der Programmcode 00, 01, 08, 07, 05 eingegeben. Damit steht der Programmcodezähler auf 5.

Nach der Eingabe des 15. hex.-dez. Zeichens wird autom. die Startfunktion ausgelöst. Werden weniger Zeichen eingegeben, muß die Eingabe mit der Starttaste abgeschlossen werden.
Die Eingabe der Nebenbedingungen nach AC 11 erfolgt prinzipiell wie die nach AC 1Ø (vergl. Abschnitt 5.5.).
Die Eingabe des Programmcodes erfolgt, wenn

a) ein neuer Programmcode eingegeben werden soll,

b) ein Netzausfall zu verzeichnen war (und damit der Datenerhalt im RAM-Speicher ausfällt), d.h. Signallampe NAS leuchtet.

Entsprechend dem Kopfprogramm erfolgt die Programmcodeeingabe stets nach Betätigung der P-Taste und der *-Taste

5.4. Ergänzungen der Makrobefehlsliste

Im Zusammenhang mit der Modulprogrammierung stehen folgende zusätzlichen Makrobefehle zur Verfügung:

ME	Modulende	FØ
MAW	Modulauswahl	F6
MSP	Modulsprung	1Ø

5.4.1. ME - Modulende

Der Befehl ME kennzeichnet das Ende eines Moduls. I teilt der Zentraleinheit mit, daß der nächste im AC 1Ø markierte Modul abgearbeitet werden soll.
Geht man von dem Beispiel des Abschnittes 5.3. aus, so steht am Programmanfang (nach der Eingabe EZ,10 und EZ,11) der Programmcodezähler auf 5. Damit wird der Modul 00 angesprochen.

Ist dieser Modul abgearbeitet (Makrobefehl ME), so wird der Programmcodezähler um 1 dekrementiert. Anschließend wird der Modul 01 adressiert und abgearbeitet. Entsteht im Programmcodezähler als Ergebnis der Dekrementierung eine 0, so wird automatisch der 1. Modul angesprochen (Im angegebenen Beispiel erfolgt nach Abarbeitung des Moduls 05 die des Moduls 00). Es ist denkbar, daß ein Modul mehrere ME-Befehle enthält.

5.4.2. MAW-Modulauswahl

Der Befehl MAW verlangt als Nebenbedingung (in AC 11) eine Zahl 1 ≤ N ≤ 15. Er wird also als Modul programmiert (mit anschließendem ME-Befehl). Er bekommt als Schlüsselzahl im AC 10 stets die Ziffer n = 9 zugeordnet.
Wird ein MAW-Befehl erkannt, so wird der Programmcodezähler auf den Wert N (aus AC 11) eingestellt und mit dem entsprechenden Modul weitergearbeitet (Programmschleifen!)

Beispiel 2:

Der eingegebene Programmcode lautet:

Programmcodezähler	9	8	7	6	5	4	3	2	1
AC 10	00,	01,	03,	06,	05,	09,	06,	03,	02,
AC 11	06,	05,	10,	07,	12,	08,	03,	10,	04,

Der Programmcodezähler wird sukzessiv dekrementiert und steht schließlich auf 4. Das bedeutet, es wird der Modul 09 abgearbeitet.
Dieser Modul soll folgendes Aussehen haben:

```
Ø9  MAW  F6
    ME   FØ
```

Nachdem MAW decodiert wurde, wird der Programmcode-zähler auf den Wert 08 eingestellt (entspricht der AC 11-Information des Moduls 09). Damit wird mit dem Modul 01 weitergearbeitet.

5.4.3. MSP-Modulsprung

An dem unter 5.4.2. beschriebenen Beispiel 2 ist ersichtlich, daß im allgemeinen keine Möglichkeit besteht, aus einer Modul-Programmschleife herauszukommen und das eingegebene Modulprogramm an anderer Stelle definiert fortzusetzen. Daher wurde der relative Sprungbefehl MSP eingeführt.
Entschlüsselt die Maschine einen MSP-Befehl, dekrementiert sie den Programmcodezähler solange, bis im AC 1∅ eine ∅9 gelesen wird. Anschließend wird der Programmcodezähler noch um eins vermindert und mit dem, in der entsprechenden AC 1∅-Tetrade markierten Modul weitergearbeitet.

Beispiel 3:

Nehmen wir den Programmcode aus Beispiel 2. Es ist vorstellbar, daß der Modul 01 folgenden Programmteil enthält:

```
        - EZ, 5
ABPT    - SPT        M01 T
        - AND 5
        .
        .
        .
M01T    - MSP
        .
        .
        .
        .
```

Somit würde im Modul 01 an der symbolischen Adresse ABFT, sofern die Eingabe EZ 5 mit der *-Taste abgeschlossen wurde, ein Sprung nach M01T erfolgen. Mit dem MSP wird der Modul verlassen. Entsprechend der obigen Darlegung würde jetzt die Maschine mit der Abarbeitung des Moduls 06 fortfahren.

5.5. Eingabe der Nebenbedingungen

Es besteht bei der Modulprogrammierung die Möglichkeit, jedem im AC 1ø markierten Modul im AC 11 eine Ziffer als Nebenbedingung zuzuordnen. Mit dieser Nebenbedingung, die wieder eine Ziffer zwischen 0 und 15 (einschließlich) sein kann, kann beispielsweise der Speicherplatz einer Operation über die Zehnertastatur des Gerätes festgelegt werden. Ferner kann die eingegebene Nebenbedingung die Kapazität bei Ausgabebefehlen, die Anzahl der Tabulatorsprünge beim TABN-Befehl oder die Einstellung des Programmcodezählers beim MAW-Befehl bestimmen.
Folgende Beispiele sollen das zeigen:

5.5.1. Nebenbedingung ACX

Alle Makrobefehle, deren Codierung laut Makrobefehlsliste in der 1. Tetrade des 1. Bytes mit ACx angegeben ist, werden normalerweise mit ACx ≠ 15 ≙ F programmiert, da AC 15 als Sonderregister definiert ist.
Bei der Modulprogrammierung kann ACx = 15 ≙ F gesetzt werden. In diesem Falle wird die dem betreffenden Modul zugeordnete Nebenbedingung als Speicherplatz angesehen.

Beispiel 4:

Programmcode	00, 07, 04, 03, 09, 04, 06
Nebenbedingung	06, 09, 00, 05, 06, 02, 03

Der Modul 07 enthält folgenden Programmteil:

```
      .
      .
      .
1000  BZ,F
      AND*   = 5 F 6 8
      SPF    1000
      .
      .
      .
      .
```

Die Nebenbedingung des Moduls 07 heißt 09. Also wird folgendes ausgeführt:

```
      .
      .
      .
1000  BZ,9
      AND  = 5 9 6 8
      SPF  1000
      .
      .
      .
```

5.5.2. Nebenbedingung Kapazität

Wird bei einem Ausgabebefehl (AND, AND*) die Kapazität 0 programmiert, so wird stattdessen wieder die Nebenbedingung des betreffenden Moduls als Ausgabekapazität gewertet.

Beispiel 5:

Programmcode	00, 02, 05, 03,
Nebenbedingung	06, 07, 02, 05,

Der Modul 00 enthält den Befehl

.
.
.
AND = 5 7 6 ∅ ,
.
.
.

so wird der Befehl

.
.
.
AND = 5 7 6 6
.
.
.

ausgeführt.

5.5.3. Nebenbedingung N-mal TAB

Hierbei verhält es sich ähnlich der Nebenbedingung KAP. N = 0 wird umgewandelt in N = Nebenbedingung aus AC 11. Es kann wieder Beispiel 5 herangezogen werden Der Modul 05 hat folgendes Aussehen:

TAB ∅ = 5 ∅ 2 ∅
ME

Daraus wird mit der Nebenbedingung 02:

TAB 2 = 5 0 2 2
ME

Somit werden in diesem Beispiel zwischen den Modules 02 und 03 zwei Tabulatorfunktionen ausgelöst.

		f	f	-	-
KO	Keine Operation. Leerbefehl	o	0	-	-
EZ	Numerische Eingabe, ER --→ ACx	e	ACx	-	-
AND	Numerische Ausgabe, ACx --→ Druck	7	ACx	PFKo	Kap.
AND*	Num.Ausg.mit Sicherheitsstern ACx --→ Druck	5	ACx	P1Ko	Kap.
ADV	Addition, ACx + MD --→ ACx	a	ACx	-	-
AD1V	Addition, ACx + 1 --→ ACx	9	ACx	-	-
SUV	Subtraktion, ACx - MD --→ ACx	b	ACx	-	-
MU (N)	Multiplikation, ACx X MD --→ MD (N=St.V.)	5	ACx	1	N
DI (N)	Division, MD : ACx --→ MD (N=St.V.)	5	ACx	0	N
VL (N)	Linksverschiebung in MD (N=Anzahl Stellen)	0	N	-	-
VR (N)	Rechtsverschiebung in MD(N=Anzahl Stellen)	1	N(N≠0)	-	-
TR	Transport, MD --→ ACx	3	ACx	-	-
TRV	Transport, ACx --→ MD	8	ACx	-	-
TRO	Einzellöschung von ACx, 0 --→ ACx	2	ACx	-	-
AMD(%)	Ausgabe Zeichen %	f	c	-	-
AMD(‰)	Ausgabe Zeichen ‰	f	d	-	-
AMD(°)	Ausgabe Zeichen °	f	e	-	-
AMD(•)	Ausgabe Zeichen •	f	5	-	-
AMD(-)	Ausgabe Zeichen - (Minus)	f	a	-	-
AMD(.)	Ausgabe Zeichen . (Punkt)	f	b	-	-
AMD(ZWR)	Ausgabe Zeichen Zwischenraum (Leerschritt)	f	1	-	-
SPU	Unbedingter Sprung	6	0(1)		
SPM	Bedingter Sprung, wenn MD < 0	6	2(3)		
SPV	Bedingter Sprung, wenn MD ≠ 0	6	4(5)	Sprung-	
SP1	Bedingter Sprung bei Sel.1	6	6(7)	adresse	
SP2	Bedingter Sprung bei Sel.2	6	8(9)		
SPT	Bedingter Sprung bei Start •	6	a(b)		
SPZ	Bedingter Sprung bei Start °	6	c(d)		
SPF	Bedingter Sprung bei Sel.F (Kap.-Überschr.)	6	e(f)		
WRZ	Wagenrücklauf mit Zeilenschaltung	f	3	-	-
FZL	Zeilenschaltung (N-zeilig)	c	N ACx	-	-
TAB	Tabulator-Sprung	f	2	-	-
TKON(K)	Konstanteneingabe, KON --→ MD (VZ-Stelle)	d	KON	-	-
AS	Adressensubstitution	f	9	-	-
ZLÖS	Löschen Selektor 1+2	f	7	-	-
ZS1	SETZEN Selektor 1	f	4	-	-
ZS2	SETZEN Selektor 2	f	8	-	-
ME	Modulende	f	0	-	-
MAW	Modulauswahl	f	6	-	-
MSP	Modulsprung	1	0	-	-
TABN	Tabulator (N= mal)	5	0	2	N
WROZ	Wagenrücklauf ohne Zeile	5	0	3	0